

REMARKS

Claims 1-30 are pending in the present application. Claims 1, 4, 6-11, 14, and 16-30 are amended for clarity. Support for amendments to claims 1, 11 and 21 may be located at least on page 10, lines 17-21. Claims 1, 4, 6-11, 14, and 16-30 are amended to provide antecedent basis. Reconsideration of the claims is respectfully requested.

I. Telephone Interview

Applicants thank Examiner Kieu D. Vu for the courtesies extended to Applicants' representative during the June 14, 2005 telephone interview. During the interview, Applicants' representative discussed amendments to the independent claims and the distinctions between the specification and the *Bereiter* and *Alexander* references. The substance of the telephone interview is summarized in the following remarks.

II. 35 U.S.C. § 103, Alleged Obviousness Based on *Bereiter* and *Alexander*

The Office Action rejects claims 1-30 under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Bereiter et al.* (U.S. Patent 5,917,492), hereinafter referred to as *Bereiter*, in view of *Alexander, III et al.* (U.S. Patent 6,002,872), hereinafter referred to as *Alexander*. This rejection is respectfully traversed.

As to independent claims 1, 11, and 21, the Office Action states:

Regarding claims 1, 11, and 21, *Bereiter* teaches displaying hierarchical dependencies comprising the steps of selecting a node from a node list displayed in one of a first and a second window region (selecting second node in window 212) and displaying one of a first node which is a subnode associated with the selected node (displaying in window 216 subnodes of selected second node). *Bereiter* differs from the claim in that *Bereiter* does not teach that the node is a routine. However, such feature is known in the art as taught by *Alexander*. *Alexander* teaches steps for monitoring performance of a program (col. 2, lines 38-40) wherein each routine in a program is represented as a node in a tree structure (col. 2, lines 55-60). It would have been obvious to one skilled in the art, having the teaching of *Bereiter* and *Alexander* before him at the time the invention was made, to use nodes to represent routines in displaying hierarchical dependencies taught by *Bereiter* so that *Bereiter's* optimizing screen estate method (col. 2, lines 20-23) can be effectively used in monitoring program performance.

Office Action dated March 15, 2005, pages 2-3.

As amended, claim 1, which is representative of the other rejected independent claims 11, and 21 with regard to similarly recited subject matter, reads as follows:

1. A method of displaying hierarchical call dependencies comprising the steps of:
selecting a routine from a routine list displayed in one of a first window region and a second window region; and
displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection. (emphasis added)

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). For an invention to be *prima facie* obvious, the prior art must teach or suggest all claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Neither *Bereiter* nor *Alexander*, taken individually or in combination, teaches or suggests “displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection,” as recited in claims 1, 11, and 21.

Bereiter is directed to a method and system for displaying a graphical representation on a display screen of a data processing system in a manner that optimizes screen “real estate” and ease of information viewing. The graphical representation may be a tree hierarchy including a plurality of expandable and collapsible nodes. The method begins as the hierarchy is displayed within a primary display window on the display screen when the user selects a node to be expanded. In response, a first display region is output within the primary display window and includes a set of one or more subnodes associated with the selected node. If a first subnode in the first display region is then selected for expansion, a second display region is then output within the first display region, and the second display region in turn includes a set of one or more subnodes of the first subnode. This generation of new “nested” display regions then continues as the user continues to explore deeper into the tree hierarchy. A given subnode display region may have one or more controls, such as a sizing button or one or more scroll bars, to facilitate display of node information within the region, and preferably each expanded node remains associated with its subnode display region using

a guideline. If generation and display of a subnode display region requires additional screen space, one or more of the subnodes in the "parent" display region are repositioned. See *Bereiter*, abstract. The Office Action states that *Bereiter* does not teach that the node is a routine. Additionally, *Bereiter* does not teach or suggest displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection, as recited in claims 1, 11, and 21. In other words, the present invention provides a method and system that allows a user to easily view the routines that are called by a selected routine in one window region and the routines that call a selected routine in another window region. Even if a routine is considered to be a node, *Bereiter* does not teach or suggest this feature. Additionally, *Bereiter* only provides an ability to display "child nodes" for a "parent node" and does not provide the ability to display "parent nodes" for a "child node". Further, displaying "child nodes" for a "parent node" is not the same as displaying a first set of routines called by said routine.

Alexander is directed to a method for monitoring performance of a program. A periodically occurring event is detected and a stack associated with the program is identified in response to a detection of the periodically occurring event, which may be a timer interrupt. The stack is examined to identify each routine that is currently executing in association with the program. Each routine is represented as a node in a tree structure. See *Alexander*, abstract. *Alexander* does not teach or suggest "displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection," as recited in claims 1, 11, and 21.

In the rejection of independent claims 1, 11, and 21, the Office Action refers to the following portions of *Alexander*:

Therefore, it would be advantageous to have an improved method and apparatus for profiling data processing systems and the applications executing within the data processing systems.

Alexander, column 2, lines 20-23.

The present invention provides a method for monitoring the performance of an entire system with one or more active programs.

Alexander, column 2, lines 38-40.

A call stack associated with the page fault identified in response to detecting the page fault. The call stack is examined to identify each routine executing when the page fault was detected. Also, each routine is represented as a node in the tree structure.

Alexander, column 2, lines 55-60.

These portions of *Alexander* do not teach or suggest “displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection,” as recited in claims 1, 11, and 21. To the contrary, these portions teach that the call stack identifies each routine executing when a page fault is detected. *Alexander* does not mention displaying a first set of routines that are called by a selected routine or a second set of routines that call a selected routine. *Alexander* only discloses that each routine that was executing when the page fault was detected is represented as a node in a tree structure.

Bereiter and *Alexander* fail to teach or suggest “displaying one of a first set of routines called by said routine in said first window region and a second set of routines calling said routine in said second window region in response to said selection.” Therefore, the alleged combination of *Bereiter* and *Alexander* does not teach or suggest these features, as recited in independent claims 1, 11, and 21.

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

One of ordinary skill in the art would not combine *Bereiter* with *Alexander* when the references are considered as a whole. In considering the references as a whole, one of ordinary skill in the art would take into account the problems recognized and solved. The present invention recognizes the problem of tracking the hierarchical chain of call dependencies of a selected routine in a software program. *Bereiter* and *Alexander* do not teach the problem or its source. *Bereiter* is directed to the problem of optimizing the

amount of the display screen that is actually being used to display information as the tree hierarchy (see *Bereiter*, column 1, lines 52-62). *Alexander* is directed toward monitoring the performance of a data processing system (see *Alexander*, column 2, lines 30-32). One of ordinary skill in the art would therefore not be motivated to combine or modify the references in the manner required to form the solution disclosed in the present invention.

Furthermore, as noted above, there is no teaching or suggestion in the references as to the desirability of including the features from the other references. *Bereiter* does not mention a desirability for monitoring performance of a data processing system and *Alexander* does not mention a desirability for displaying a tree structure in a graphical user interface. As the Examiner has failed to demonstrate any motivation or incentive in the prior art to combine and modify the references so as to achieve the claimed invention, the alleged combination can only be the result of impermissible hindsight reconstruction using Applicants' own disclosure as a guide. While Applicants understand that all examination entails some measure of hindsight, when the rejection is based completely on hindsight, as in the present case, to the exclusion of what can be gleaned from the references, then the rejection is improper and should be withdrawn.

Further, the present invention provides a method of displaying hierarchical call dependencies so that a programmer may have knowledge of the calling and called routines that may be affected if changes are applied to any given routine (see Specification, page 2, lines 5-16). The combination of *Bereiter* and *Alexander*, even assuming it is proper, fails to provide this feature since *Alexander* is concerned with the performance of a program when executing and *Bereiter* is concerned with displaying a tree structure. *Bereiter* and *Alexander* are not concerned with the calling dependencies of a program.

Therefore, *Bereiter* and *Alexander*, taken individually or in combination, do not teach or suggest the features of independent claims 1, 11 and 21, and Applicants respectfully request withdrawal of the rejection under 35 U.S.C. § 103(a) with respect to these claims.

Dependent claims 2-10, 12-20 and 22-30 are also not obvious over *Bereiter* and *Alexander*, taken individually or in combination, at least by virtue of their dependency on independent claims 1, 11 and 21, respectively. Accordingly, Applicants also respectfully

request withdrawal of the rejection of claims 2-10, 12-20 and 22-30 under 35 U.S.C. § 103(a).

In addition, with regard to claims 2, 12 and 22, *Bereiter* and *Alexander*, either taken alone or in combination, do not teach or suggest that a first window region comprises a calls window region and a second window region comprises a called-by window region. *Bereiter* and *Alexander* do not mention a calls window region or a called-by window region. *Bereiter* actually teaches away from this feature since each group of subnodes is displayed in a new window to optimize screen real estate for viewing the subnodes. Thus, *Bereiter* and *Alexander* do not teach or suggest the features as described in claims 2, 12, and 22.

With regard to claims 3, 13 and 23, *Bereiter* and *Alexander*, either taken alone or in combination, do not teach or suggest that the routine list is contained in a plurality of data structures stored in a database. The cited portion of *Bereiter* only states that the view may represent a network of machines, a website, a database, a file system, or any other hierarchical set of data. Similarly, *Bereiter* and *Alexander* do not teach the specific features of claims 6-8, 16-18, and 26-28 with regard to accessing a data structure stored in a database.

With regard to claims 10, 20 and 30, *Bereiter* and *Alexander*, either taken alone or in combination, do not teach or suggest displaying said routine list in said first window region and said second window region. In other words, the same routine list is displayed in both the first window region and the second window region. This provides the ability for a user to view both a first set of routines that are called by a selected routine and a second set of routines that call a selected routine at the same time. *Bereiter* and *Alexander*, either taken alone or in combination, do not teach or suggest this feature.


Thus, in addition to being dependent on their respective independent claims, claims 2-10, 12-20, and 22-30 are also distinguished over the *Bereiter* and *Alexander* references based on the specific features recited therein.

III. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: June 15, 2005

Respectfully submitted,



Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/vja